A4] Imagine a publishing company which does marketing for book and audiocassette versions. Create a class publication that stores the title (a string) and price (type float) of a publication.From this class derive two classes: book, which adds a page count(type int), and tape, which adds a playing time in minutes(type float). Write a program that instantiates the book and tape classes,allows user to enter data and displays the data members.If an exception is caught, replace all the data member values with zero values. Link\*\*\* <https://www.programiz.com/online-compiler/6eY2FYwxM83QT>

Without comments \*\*\* https://www.programiz.com/online-compiler/1hITYM8iUqtwf

#include <iostream>

#include <string>

using namespace std;

// Base class: publication

class publication {

private:

string title;

float price;

public:

// Method to input title and price

void add() {

cout << "Enter Title of the Publication: ";

cin.ignore(); // Clear the input buffer

getline(cin, title);

cout << "Enter Price of Publication: ";

cin >> price;

// Validate price

if (price <= 0) {

throw "Invalid price!";

}

}

// Method to display title and price

void display() const {

cout << "Title: " << title << ", Price: $" << price;

}

};

// Derived class: book

class book : public publication {

private:

int page\_count;

public:

// Method to input book data

void add\_book() {

try {

add(); // Input common data (title, price)

cout << "Enter Page Count of Book: ";

cin >> page\_count;

if (page\_count <= 0) {

throw page\_count;

}

}

catch (...) {

cout << "Invalid data entered! Resetting to zero values." << endl;

page\_count = 0;

}

}

// Method to display book data

void display\_book() const {

display(); // Display common data (title, price)

cout << ", Page Count: " << page\_count << endl;

}

};

// Derived class: tape

class tape : public publication {

private:

float play\_time;

public:

// Method to input tape data

void add\_tape() {

try {

add(); // Input common data (title, price)

cout << "Enter Play Duration of the Tape (in minutes): ";

cin >> play\_time;

if (play\_time <= 0) {

throw play\_time;

}

}

catch (...) {

cout << "Invalid data entered! Resetting to zero values." << endl;

play\_time = 0;

}

}

// Method to display tape data

void display\_tape() const {

display(); // Display common data (title, price)

cout << ", Play Time: " << play\_time << " min" << endl;

}

};

int main() {

book b1; // Single book object

tape t1; // Single tape object

// Add and display a book

cout << "Enter Book Information:\n";

b1.add\_book();

cout << "\nBook Information:\n";

b1.display\_book();

// Add and display a tape

cout << "\nEnter Tape Information:\n";

t1.add\_tape();

cout << "\nTape Information:\n";

t1.display\_tape();

return 0;

}